CS 467 Capstone
Fall 2020

Mid-Point Project Check

Music Visualization Application Team

Samnang Penh
John Sy

Submission Contents

Included in Mid-Point Project Check:

- Mid-Point Write-Up
- Zip file containing source code (Github)
- Video instructions: https://media.oregonstate.edu/media/t/1_q7fl0laz
- URL to the application: http://adrift108.pythonanywhere.com/

Project Status

We are presently past the halfway point and are currently working on the 3D animations/particle effects that will be directly informed by the music playing. We have a functional web application that can play music provided by the user via url from SoundCloud and display a test animation that is independent of the music. However, our application can only play individual tracks and not tracks from a playlist at the moment. The user interface for the music is visually and functionally complete. The music player can take any valid SoundCloud track url that allows for streaming and subsequently play that track with the track details updated on the player. If the link is invalid for any reason, the user will be notified with an error message. The application has also been tested for major bugs.

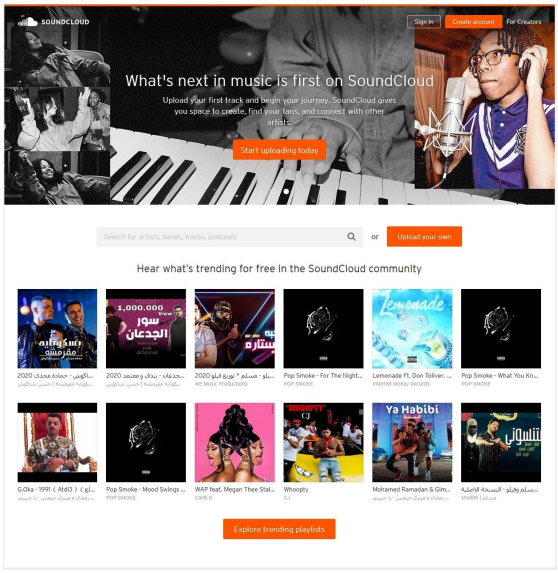
We have completed all our objectives from the project plan on schedule up to this point with no issues. The team's upcoming tasks include processing tracks from playlists, creating original Three.js animations that coincide with the music, as well as reorganization of the existing code structure for modularity. Additionally, we are in the process of researching and implementing performance improvements for our application.

User Instructions

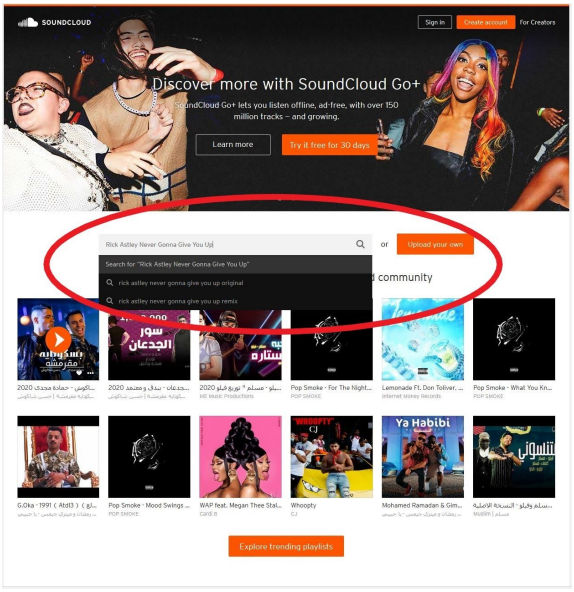
Instructions will consist of two sections. The application uses the SoundCloud API, so valid inputs to the application are SoundCloud track urls. SoundCloud is one of the most popular music distribution platforms for both artists and listeners that allows people to stream, download, or publish music. The first section will go over how to navigate SoundCloud to find a valid track to use in the application. The second section will go over how to use the application itself once you have a SoundCloud track link.

SoundCloud Section

1) Navigate to SoundCloud using a web browser at https://soundcloud.com/.
The home page as of the date of this document will appear as below:

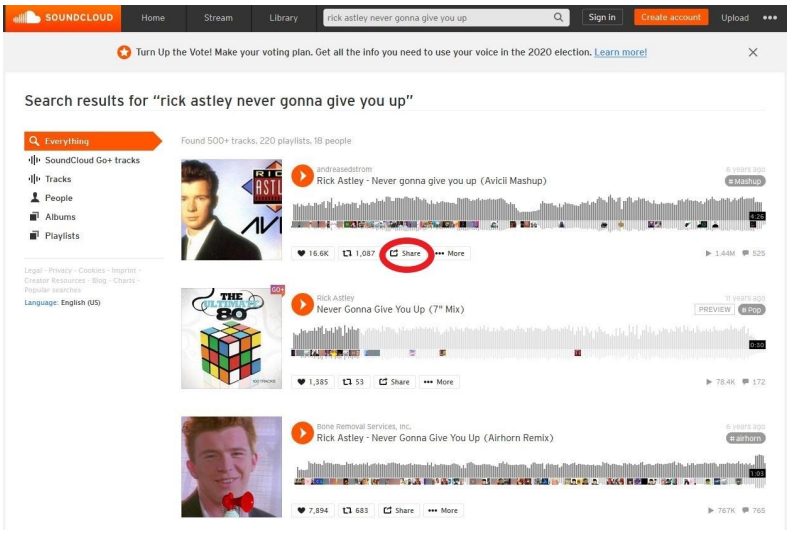


2) Look at the middle of the homepage where you should see a search bar as circled in red in the screenshot below. Search for a specific song using the title of the song, and/or the artist.

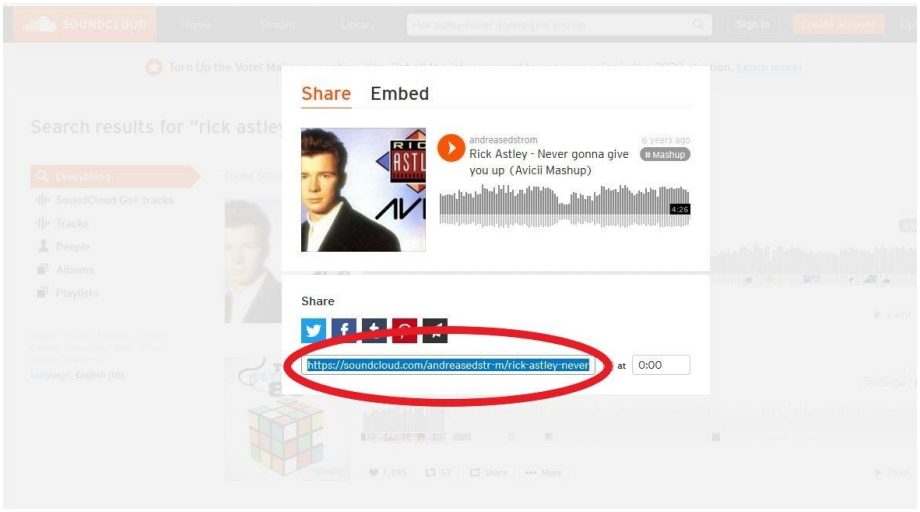


3) Once you have found a song based on search results or your own discovery, click the "share" button on any given track, as circled below. **Note that the application currently works on individual tracks, not tracks that are part of a larger playlist. Be sure you are saving the url of a single SoundCloud track and not a playlist (tracks that are a part of a playlist contain the word "set" in the url).**

If you do not wish to search for a track url, feel free to use https://soundcloud.com/lemonade_psycho/never-gonna-give-you-up-lofi-remix and save it to your clipboard, then skip to the Music Visualization Application Section.



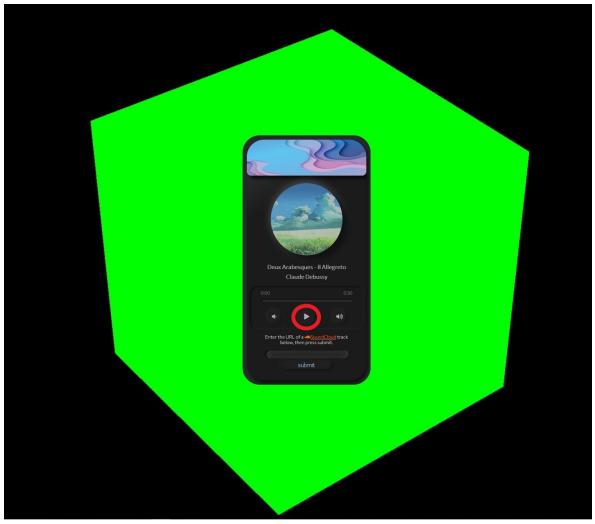
4) A popup will appear with the url of the SoundCloud Track as highlighted below. Copy that url to your clipboard. Save this url for when you interact with the main Music Visualization Application.



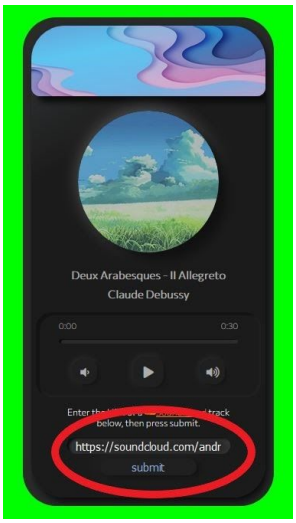
Music Visualization Application Section

1) Navigate to the application using a web browser at http://adrift108.pythonanywhere.com/.

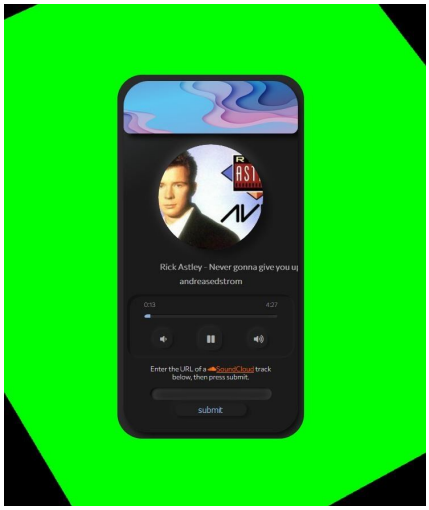
2) To play the default song, press the middle Play button in the music control panel as circled in red below. The buttons to the left and right of the middle Play button control Volume Down and Volume Up functions respectively.



3) To play the song you have selected from the SoundCloud section of the instructions prior, paste the URL into the text form at the bottom of the music player interface, followed by pressing the "submit" button as circled in red below.



4) If the application is successful in retrieving the SoundCloud track, the track information will update on the Music Player interface (if the track could not be loaded, skip to Step 5). Please note that loading the song will take a few seconds, indicated by the loading icon on the middle button. Once the song has fully loaded, the loading icon will change to a Play icon. Pressing Play will then begin playback of the track. Use the middle button once again to Play and Pause the track once playback has begun.



5) If the SoundCloud track url you provided was either not valid, or the track is not available for streaming, an error message will appear at the bottom of the player. Navigate to the SoundCloud website and try a different track/song url to submit as noted in the SoundCloud section. If you need to return to SoundCloud for any reason, click the SoundCloud link in orange above the text/url input form.

Application Objectives
To stream a user-selected music track from SoundCloud and play it through the music visualization application while an animation is playing independent of the music in the background.

What to Look For:
After submitting a valid SoundCloud track url and clicking the play button after the track has loaded, the track should be played and outputted to the browser while an animation plays in the background.

Application States
The application will be in one of the below states at any time, and each state will have visual indications. Presently, for debugging purposes, a test animation will play in the background regardless of which state the application is in. However, once an animation is finalized and implemented, the animation will sync with the various states.

Initial State: The application enters an *Initial State* on the first visit to the application, and a default classical music track is loaded by default and then the application enters a *Paused State*. However, the user is encouraged to load a SoundCloud music track.

Loading State: The application enters a *Loading State* once a user submits a link for a SoundCloud music track and the application is processing the request. In this state, the play/pause button converts to a rotating double arrow and all functionality of the application is temporarily paused while it gathers the necessary data to stream the music track and display track information.

Playing State: The application enters a *Playing State* once the user clicks on the play button from a *Paused State*. In this state, the play button converts to a pause button and the input music track url passed to the application is played, progressed, and outputted to the browser.

Paused State: The application enters a *Paused State* once the user clicks on the pause button from a *Playing State*. In this state, the pause button converts to a play button and the music track is no longer outputted to the browser and progress of the track is halted.

End State: The application enters an *End State* once the music track has completed playing. In this state, the application sets the progress of the track to the beginning, resets the progress bar, and then enters a *Paused State*.

Error State: The application enters an *Error State* if any issues are encountered. Any errors will be displayed to the user via the user interface.

References

https://developers.soundcloud.com/docs/api/reference
https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API
https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API
https://javascript.info/async-await
https://threejs.org/docs
https://discoverthreejs.com/book/contents/
https://threejsfundamentals.org/
https://github.com/goldfire/howler.js#documentation
https://www.freepik.com/free-vector/wavy-abstract-background_3708084.htm
https://archive.org/details/cd_debussy-piano-works